LINUX NETWORKING

The OSI (Open Systems Interconnection) model is a conceptual framework used to understand network interactions in seven layers, each representing specific functions within networking. Here's an explanation of the OSI model with simplified text diagrams for each layer.

OSI Model Layers

| Layer 7 - Application | Application Layer

| Layer 6 - Presentation | Presentation Layer

| Layer 5 - Session | Session Layer

| Layer 4 - Transport | Transport Layer

| Layer 3 - Network | Network Layer

| Layer 2 - Data Link | Data Link Layer

| Layer 1 - Physical | Physical Layer

Layer 1 - Physical Layer

Handles the actual physical connection between devices, defining how bits are transmitted over physical media (e.g., cables, radio waves).

--- Physical Media --->

| Bits: 0s and 1s |

- **Example:** Ethernet cables, fiber optics.

Layer 2 - Data Link Layer

Establishes a reliable link between directly connected nodes and handles error correction from the Physical layer.

| Frame Header | Data | Frame Trailer |

- **Protocols:** Ethernet, Wi-Fi (IEEE 802.3, IEEE 802.11)
- **Devices:** Switches, Bridges.

Layer 3 - Network Layer

Responsible for logical addressing and path determination, enabling data to travel from source to destination.

| IP Header | Frame | IP Trailer |

- **Protocols:** IP (Internet Protocol), ICMP.
- **Devices:** Routers.

Layer 4 - Transport Layer

Ensures data transfer reliability and controls flow; can establish connections and manage error correction.

| TCP/UDP Header | IP Packet |

- **Protocols:** TCP (Transmission Control Protocol), UDP (User Datagram Protocol).
- **Function:** Port numbers, flow control, and segmentation.

Layer 5 - Session Layer

Manages sessions or connections between applications, handling the setup, management, and termination of connections.

| Session Header | Transport Segment |

- **Example:** Controls communication between different web servers and browsers.

Layer 6 - Presentation Layer

Translates data formats, ensuring interoperability between different data encoding systems.

| Data Encryption/Decryption | Application Data |

- **Functions:** Encryption, compression, data conversion.

Layer 7 - Application Layer

Closest to the end user, this layer facilitates network access for applications.

| Application Header | Encrypted Data |

- **Examples:** HTTP, FTP, SMTP.
- **Function:** Enables web browsing, email services, file transfers.

Putting It All Together: Data Flow Example

Here's how data would flow through these layers in a real-world example, such as browsing a website:

1. **Application Layer (Layer 7):** User requests a webpage. HTTP protocol is used.
2. **Presentation Layer (Layer 6):** Data is encrypted or compressed as necessary.

3. **Session Layer (Layer 5):** Session is established with the server for the web request.
4. **Transport Layer (Layer 4):** Data is segmented, and port numbers are applied (e.g., HTTP uses port 80).
5. **Network Layer (Layer 3):** The IP address of the server is assigned for routing.
6. **Data Link Layer (Layer 2):** Data is framed for local network transmission.
7. **Physical Layer (Layer 1):** Data is transmitted over the physical medium (e.g., Wi-Fi or Ethernet) to the server.

Each layer in the OSI model relies on the layer below it to deliver its data and provides services to the layer above it. Understanding these layers helps in diagnosing network issues, as each layer has specific responsibilities and potential failure points.

IP Addressing: IPv4 vs. IPv6 in Detail

IPv4

IPv4 addresses are written in **dotted decimal notation**, with four sections separated by dots, each representing an **octet** (8 bits) of the 32-bit address. Each octet can range from 0 to 255.

- **Example**: 192.168.1.10
- **Binary Form**: 11000000.10101000.00000001.00001010

IPv4 Address Classes

IPv4 addresses are divided into classes based on the first octet:

- **Class A**: Starts from 1.0.0.0 to 126.255.255.255 with a default subnet mask of /8 (255.0.0.0).
- **Class B**: Starts from 128.0.0.0 to 191.255.255.255 with a default subnet mask of /16 (255.255.0.0).
- **Class C**: Starts from 192.0.0.0 to 223.255.255.255 with a default subnet mask of /24 (255.255.255.0).

Example of Using IPv4 in Subnetting:

For an IP like 192.168.1.0/24, we know:

- **Network Address**: 192.168.1.0
- **Broadcast Address**: 192.168.1.255
- **Range for Host IPs**: 192.168.1.1 to 192.168.1.254

IPv6

IPv6 uses **128 bits**, written in **hexadecimal notation** separated by colons. IPv6 provides a massive address space to overcome the limitations of IPv4.

- **Example**: 2001:0db8:85a3:0000:0000:8a2e:0370:7334
    - **Abbreviated Form**: 2001:db8:85a3::8a2e:370:7334 (using :: to represent consecutive zeroes)

Configuring IP Addresses on Linux

Static IP Configuration Using Network Configuration Files

On most Linux distributions, you configure static IPs in network configuration files.

1. **Debian-based Systems** (e.g., Ubuntu):
    - File: /etc/network/interfaces

auto eth0

iface eth0 inet static

address 192.168.1.10

netmask 255.255.255.0

gateway 192.168.1.1

2. **RedHat-based Systems** (e.g., CentOS):
    - File: /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=static

IPADDR=192.168.1.10

NETMASK=255.255.255.0

GATEWAY=192.168.1.1

ONBOOT=yes

Dynamic IP Configuration Using dhclient

Using dhclient allows you to assign an IP via DHCP:

sudo dhclient eth0

The above command requests an IP from the DHCP server for eth0.

Subnetting in Detail

Subnetting divides a network into smaller subnetworks. This is essential for network organization, security, and efficient use of IP addresses.

Subnet Mask

A subnet mask specifies how much of the IP address represents the network vs. the host. For example, in 192.168.1.10 with subnet mask 255.255.255.0, the **first three octets (192.168.1)** represent the network, while the **last octet** (.10) represents the host.

Example: Subnetting a Network

Consider a network 192.168.1.0/24, which allows 256 IPs (254 usable IPs for hosts).

To create **four smaller subnets**, you would use a /26 subnet mask (255.255.255.192), providing 64 IPs per subnet.

Subnet	Range	Usable IP Range
/26	192.168.1.0 - 192.168.1.63	192.168.1.1 - 192.168.1.62
/26	192.168.1.64 - 192.168.1.127	192.168.1.65 - 192.168.1.126
/26	192.168.1.128 - 192.168.1.191	192.168.1.129 - 192.168.1.190
/26	192.168.1.192 - 192.168.1.255	192.168.1.193 - 192.168.1.254

CIDR Notation in Detail

Classless Inter-Domain Routing (CIDR) notation is a compact representation of IP addresses and subnet masks.

- **CIDR Notation**: /24 means 24 bits for the network portion.
- **Benefits**: Simplifies subnetting and allows for efficient address allocation.

Example Calculations with CIDR

1. **192.168.10.0/24**:
    - **Subnet Mask**: 255.255.255.0
    - **Network Portion**: 24 bits
    - **Host Portion**: 8 bits
    - **Total IPs**: 28=2562^8 = 25628=256 (254 usable IPs)
2. **192.168.10.0/26**:
    - **Subnet Mask**: 255.255.255.192
    - **Network Portion**: 26 bits
    - **Host Portion**: 6 bits
    - **Total IPs**: 26=642^6 = 6426=64 (62 usable IPs)

Real-World CIDR Application Example

Imagine configuring subnets for a large office network with different departments. You might assign:

- **HR Department**: 192.168.1.0/26
- **IT Department**: 192.168.1.64/26
- **Sales Department**: 192.168.1.128/26 Each department would have its range of IPs while remaining within the main 192.168.1.0/24 network.

Configuring and Verifying IP and Subnet on Linux

Adding an IP Address with CIDR Notation

Using ip:

sudo ip addr add 192.168.1.10/24 dev eth0

This assigns the IP 192.168.1.10 with a subnet mask /24 to eth0.

Viewing Subnet Information

You can see your network configuration with:

ip addr show eth0

This command displays the assigned IP, CIDR notation, and other details.

Advanced Example: Custom Subnetting and CIDR Calculation

Let's subnet a Class B network, 172.16.0.0/16, into multiple smaller networks with a /20 mask:

Subnet	Network Address	Range Start	Range End	CIDR Notation
First Subnet	172.16.0.0	172.16.0.1	172.16.15.254	172.16.0.0/20
Second Subnet	172.16.16.0	172.16.16.1	172.16.31.254	172.16.16.0/20
Third Subnet	172.16.32.0	172.16.32.1	172.16.47.254	172.16.32.0/20

Each /20 subnet provides 4096 addresses (4094 usable IPs).

What is eth0?

In a Linux system, eth0 is the name given to the computer's first wired network connection. Think of it as the main "port" where you plug in a network cable to get internet or connect to other computers in a network.

Here's an analogy:

- Imagine your computer is a house, and the network is like a series of pipes connecting houses together.
- eth0 would be the main pipe that brings internet or data into your house. It's the default pathway for sending and receiving data if you're using a physical network cable.

Why is it Called eth0?

- eth stands for "Ethernet," which is a type of wired network connection.
- The 0 means it's the first Ethernet connection detected by the computer. If you added a second Ethernet connection (like plugging in a second network card), it would likely be called eth1.

So eth0 is simply the name Linux gives to the primary Ethernet connection on your computer.

What Do We Use eth0 For?

When you connect your computer to a network using an Ethernet cable, you're using eth0. This connection allows you to:

- **Get an IP address from your router so you can access the internet.**
- **Communicate with other computers on your local network.**

Key Networking Concepts in Linux

1. **Network Interfaces:** A network interface is a point of interaction between a system and its network. It could be physical (e.g., Ethernet card) or virtual (e.g., loopback).
2. **IP Addressing:** Every device on a network has an IP address, allowing it to communicate with other devices. It could be IPv4 or IPv6.
3. **Routing:** Routing is determining the path that data packets take from the source to the destination.
4. **DNS (Domain Name System):** DNS resolves human-readable domain names (e.g., google.com) to IP addresses.
5. **ARP (Address Resolution Protocol):** Maps IP addresses to MAC addresses so that packets can be sent to the correct hardware address on the LAN.

1. ip Command

The ip command is used for managing network interfaces, routes, and other network settings.

Example 1: Viewing All Network Interfaces and IP Addresses

ip addr show

Output:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::5054:ff:fe12:3456/64 scope link
```

Explanation:

- lo is the loopback interface, often used for local communication within the same machine.
- eth0 is an Ethernet interface with an IPv4 address 192.168.1.10/24 and an IPv6 address fe80::5054:ff:fe12:3456/64.
- The UP state indicates that the interface is active.

Example 2: Adding an IP Address to an Interface

```
sudo ip addr add 192.168.1.20/24 dev eth0
```

This command assigns the IP 192.168.1.20 to the eth0 interface. It's useful when configuring multiple IP addresses on a single interface.

---

2. ifconfig Command

ifconfig is an older tool but still useful for configuring network interfaces.

Example: Bringing an Interface Up

```
sudo ifconfig eth0 up
```

Explanation: This command activates the eth0 interface. If an interface is down, it won't be able to send or receive network traffic.

---

3. ping Command

The ping command checks connectivity by sending ICMP packets to a remote host.

Example: Pinging a Domain

```
ping google.com
```

Output:

```
PING google.com (142.250.74.206): 56 data bytes
64 bytes from 142.250.74.206: icmp_seq=0 ttl=115 time=10.2 ms
64 bytes from 142.250.74.206: icmp_seq=1 ttl=115 time=10.1 ms
```

Explanation:

- Each line shows a response from Google's server with round-trip times (time=10.2 ms).
- This indicates that there is connectivity between your system and the remote server.

---

4. traceroute Command

traceroute identifies the path packets take to reach a destination, showing each router along the way.

Example: Tracing the Route to a Domain

```
traceroute google.com
```

Output:

```
1  192.168.1.1 (192.168.1.1)  2.3 ms  1.6 ms  1.7 ms
2  10.0.0.1 (10.0.0.1)  3.2 ms  4.0 ms  4.5 ms
3  172.217.15.78 (172.217.15.78)  15.0 ms  14.7 ms  14.8 ms
```

Explanation:

- Each line represents a "hop," or router, along the path to the destination.
- The times (2.3 ms, etc.) indicate how long it took for packets to travel to each hop.

---

5. netstat Command

netstat shows network connections, routing tables, and interface statistics.

Example: Listing All Listening Ports

```
netstat -tuln
```

Output:

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:68              0.0.0.0:*
```

Explanation:

- tcp and udp show the protocols in use.
- The port 22 is open and listening for SSH connections.
- The LISTEN state indicates the server is waiting for incoming connections.

---

6. ss Command

ss is a newer, faster alternative to netstat, offering detailed socket statistics.

Example: Showing All Listening Sockets

```
ss -tuln
```

Output:

```
Netid  State      Recv-Q Send-Q  Local Address:Port   Peer Address:Port
tcp    LISTEN     0      128     0.0.0.0:22          0.0.0.0:*
udp    UNCONN     0      0       0.0.0.0:68          0.0.0.0:*
```

Explanation:

- This output is similar to netstat -tuln, showing open ports and their states.
- Netid indicates the protocol used (tcp, udp).
- Recv-Q and Send-Q show queue sizes for receiving and sending data.

---

7. route Command

route allows you to view and manipulate the routing table.

Example: Viewing the Routing Table

```
route -n
```

Output:

Kernel IP routing table

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    100    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
```

Explanation:

- 0.0.0.0 represents the default route, directing traffic to 192.168.1.1 (gateway).
- UG means this is a gateway (G) route that's up (U).
- The routing table helps Linux decide which gateway to use when sending packets.

---

8. dig Command

dig queries DNS servers for information about a domain.

Example: Querying an A Record

```
dig google.com
```

Output:

```
;; ANSWER SECTION:
google.com.          300    IN     A      142.250.74.206
```

Explanation:

- The A record maps google.com to an IP address 142.250.74.206.
- The 300 indicates the time-to-live (TTL) in seconds, specifying how long this record is cached.

---

9. iptables Command

iptables is a firewall tool for managing packet filtering rules.

Example: Allowing SSH Connections

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Explanation:

- This command allows incoming connections on TCP port 22 (SSH).
- -A INPUT adds a rule to the INPUT chain.
- -j ACCEPT tells iptables to accept matching packets.

---

10. hostname Command

hostname displays or sets the system's hostname.

Example: Displaying the Current Hostname

```
hostname
```

Output:

```
my-linux-server
```

Explanation:

- The command simply outputs the current hostname of the system, which is useful for identification.

---

11. nmcli Command

nmcli is a command-line interface for NetworkManager, used to manage network connections.

Example: Listing All Connections

nmcli con show

Output:

```
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  1f926ca4-7ed1-4f82-9f87-1e2d0a1e3fa3  ethernet  eth0
```

Explanation:

- nmcli con show lists all active network connections and their UUIDs.
- eth0 is the device associated with this connection.

---

12. tcpdump Command

tcpdump captures and analyzes packets on a network interface.

Example: Capturing Packets on eth0

sudo tcpdump -i eth0

Output:

14:20:01.123456 IP 192.168.1.10.12345 > 142.250.74.206.http: Flags [S], seq 1234567890, win 65535, options [mss 1460,sackOK,TS val 123456789 ecr 0,nop,wscale 6], length 0

Explanation:

- tcpdump captures packets on eth0, displaying source and destination IPs, ports, and packet flags.
- Here, 192.168.1.10 is sending a SYN packet to 142.250.74.206 on HTTP port 80.

Summary Table with Explanation

Command	Example	Explanation
ip	ip addr show	Shows all network interfaces and their IP addresses.
ifconfig	sudo ifconfig eth0 up	Activates the eth0 interface, allowing it to send/receive data.
ping	ping google.com	Tests connectivity to google.com by sending ICMP packets.

Command	Example	Explanation
traceroute	traceroute google.com	Traces the route packets take to google.com, showing each hop.
netstat	netstat -tuln	Lists all open ports and listening services.
ss	ss -tuln	Displays active listening sockets, like netstat, but faster.
route	route -n	Shows the routing table, including the default gateway.
dig	dig google.com	Resolves google.com to its IP address by querying DNS.
iptables	sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT	Allows incoming SSH connections on port 22.
hostname	hostname	Displays the current hostname of the system.
nmcli	nmcli con show	Lists all active network connections and associated devices.
tcpdump	sudo tcpdump -i eth0	Captures and displays packets on eth0, useful for network diagnostics.

These commands cover various aspects of network configuration, monitoring, and troubleshooting in Linux, providing powerful tools for effective network management.